

Creamos una aplicación con nest en npm

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

rodri@Seraph-PK MINGW64 ~/OneDrive/Escritorio/EXAMEN TALLER
$ nest new app1
⚡ We will scaffold your app in a few seconds..

? Which package manager would you ❤️ to use? npm
CREATE app1/.eslintrc.js (688 bytes)
CREATE app1/.prettierrc (54 bytes)
CREATE app1/nest-cli.json (179 bytes)
CREATE app1/package.json (2812 bytes)
CREATE app1/README.md (3413 bytes)
CREATE app1/tsconfig.build.json (101 bytes)
CREATE app1/tsconfig.json (567 bytes)
```

Creamos el controlador

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

to help us maintain this package.

👉 Donate: https://opencollective.com/nest

rodri@Seraph-PK MINGW64 ~/OneDrive/Escritorio/EXAMEN TALLER
$ nest generate controller index
CREATE index/index.controller.ts (103 bytes)
CREATE index/index.controller.spec.ts (503 bytes)

rodri@Seraph-PK MINGW64 ~/OneDrive/Escritorio/EXAMEN TALLER
$
```

Instalamos ejs

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

CREATE index/index.controller.ts (103 bytes)
CREATE index/index.controller.spec.ts (503 bytes)

rodri@Seraph-PK MINGW64 ~/OneDrive/Escritorio/EXAMEN TALLER
$ npm install ejs

added 16 packages in 3s

2 packages are looking for funding
  run `npm fund` for details

rodri@Seraph-PK MINGW64 ~/OneDrive/Escritorio/EXAMEN TALLER
$
```

Creamos la carpeta de vistas

```
rodri@Seraph-PK MINGW64 ~/OneDrive/Escritorio/EXAMEN TALLER
$ mkdir views

rodri@Seraph-PK MINGW64 ~/OneDrive/Escritorio/EXAMEN TALLER
$
```

Creamos un html con el nombre de nuestro controlador, que es index

Dentro de main.ts configuramos las plantillas para html

```
app1 > src > TS main.ts > ...
1 import { NestFactory } from '@nestjs/core';
2 import { AppModule } from './app.module';
3 import { join } from 'path';
4 import * as ejs from 'ejs';
5 import { NestExpressApplication } from '@nestjs/platform-express';
6
7 async function bootstrap() {
8
9   const app = await NestFactory.create<NestExpressApplication>(AppModule);
10
11   app.engine('html', ejs.renderfile);
12   app.set('view engine', 'html');
13   app.set('views', join(__dirname, '..', 'views'));
14
15   await app.listen(3000);
16
17 }
18 bootstrap();
19 |
```

Definimos rutas en el controlador

```
index > TS index.controller.ts > ...
1 import { Controller, Get, Render } from '@nestjs/common';
2
3 @Controller('index')
4 export class IndexController {
5
6     @Get()
7     @Render('index')
8     index(){
9         return
10     }
11
12 }
13
```

Añadimos una imagen y un fondo

```
1 <html>
2   <head>
3     <title>Index controller</title>
4   </head>
5   <style>
6     body{
7       background-color: #brown;
8       display: flex;
9     }
10    .imgz{
11      margin: 0 auto;
12      width: 50%;
13      height: 50%;
14    }
15  </style>
16  <body>
17    
18    <p>Puro Pollito</p>
19  </body>
20 </html>
```

Iniciamos la aplicación

```
rodri@Seraph-PK MINGW64 ~/OneDrive/Escritorio/EXAMEN TALLER/app1 (master)
$ nest start
[Nest] 18576 - 02/21/2024, 5:31:20 PM LOG [NestFactory] Starting Nest application...
[Nest] 18576 - 02/21/2024, 5:31:20 PM LOG [InstanceLoader] AppModule dependencies initialized +6ms
[Nest] 18576 - 02/21/2024, 5:31:20 PM LOG [RoutesResolver] AppController {/}: +15ms
[Nest] 18576 - 02/21/2024, 5:31:20 PM LOG [RouterExplorer] Mapped {/, GET} route +2ms
[Nest] 18576 - 02/21/2024, 5:31:20 PM LOG [NestApplication] Nest application successfully started +3ms
```

Ingresamos al link de nuestro localhost

